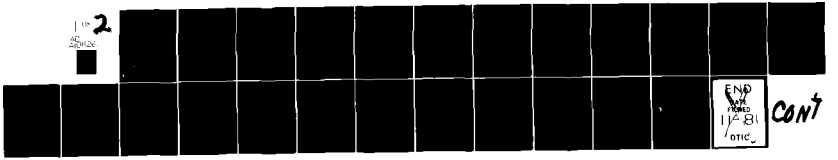
AD-A106 261 NAVAL SURFACE WEAPONS CENTER SILVER SPRING MD F/G 9/2
THE SAFEGUARDS AUTOMATED FACILITY EVALUATION (SAFE) ON THE INTE--ETC(U)
AUG 81 M S SCHWARTZ
UNCLASSIFIED NSWC/TR-81-321 NL



CONT

NSWC TR 81-321

THE SAFEGUARDS AUTOMATED FACILITY EVALUATION (SAFE) ON THE INTERDATA 7/32

BY MELANIE S. SCHWARTZ

WEAPONS SYSTEMS DEPARTMENT

AUGUST 1981

NAVAL SURFACE WEAPONS CENTER

Dahlgren, Virginia 22448 • Silver Spring, Maryland 20910

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: NSWC TR-81-321	2. GOVT ACCESSION NO.: AD-A106 361	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): THE SAFEGUARDS AUTOMATED FACILITY EVALUATION (SAFE) ON THE INTERDATA 7/32.	5. TYPE OF REPORT & PERIOD COVERED: Final
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Melanie S. Schwartz	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Surface Weapons Center, White Oak, Silver Spring, Maryland 20910	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 63571N, S0812SL, S0812L001, N78
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE: August 1981
	13. NUMBER OF PAGES: 26
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approve for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

SAFE
Free Format
Unformatted
Graphic Display
Changes

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the conversion process that was necessary for the Safeguards Automated Facility Evaluation (SAFE) to run on the Interdata 7/32. It also describes differences in the operation of the model by pointing out changes to the way Volume III of the SAFE Users Manual is to be used for the Interdata version.

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE

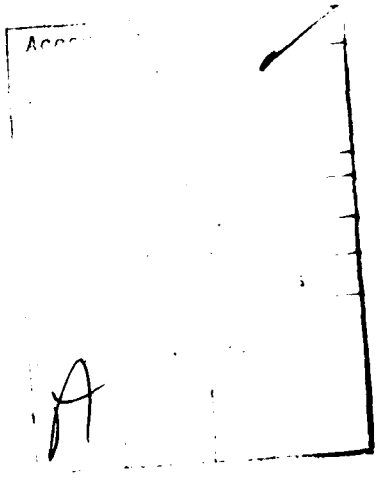
UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FOREWORD

The Safeguards Automated Facility Evaluation (SAFE) model is a collection of programs developed at Sandia Laboratories in Albuquerque, New Mexico for the CDC 6600 computer. The programs were converted for execution on an Interdata 7/32 computer for the Shipboard Nuclear Weapons Security program. This report describes this effort.

JOHN M. WACK
By direction



CONTENTS

ILLUSTRATIONS

Chapter 1

CONVERSION OF SAFE FROM THE CDC 6600 TO THE INTERDATA 7/32

PROCEDURE DIFFERENCES

The Safeguards Automated Facility Evaluation (SAFE) is divided up into three sets of programs: UAREA, UNPREP, and SAFE. The first two are actually pre-processors which transform the input data into a format which the first of the SAFE programs can recognize. However, UNPREP also contains a section which calculates the dimensions for a set of arrays present in most of the SAFE programs. This implies that before being run, the SAFE programs must be compiled with these new dimensions.

The process for doing this was designed for a CDC machine and had to be modified to reflect the needs of the Interdata. The CDC version generates an editor command file in UNPREP which operates on a generic source version of SAFE, changing dimensions from symbolic names to constants, and saving the programs individually under their appropriate names. Since only source code is saved from one time to the next, the programs are re-compiled before they are run, and the new versions will be used on executions to follow. The process actually becomes somewhat simplified on the Interdata. Since executable task images are saved on the disk from one time to the next, time-consuming compilation and task establishment do not have to be done before every execution. The dimensions can be controlled by including PARAMETER statements in the generic programs which associate the symbolic names with constants. These statements may be kept in a separate file and included in the programs using $INCLUDE, so that only that file need ever be edited.

The file must be edited and re-compilation and task establishment performed only when current dimensions exceed previous ones. This entire process is contained within UNPREP. It begins by running REGION, which in addition to generating an output file for SAFE to use, generates a file called NPARAMS containing the PARAMETER statements needed to adequately dimension the SAFE arrays. Next, the current PARAMETER statement file, PARAMS, is copied to PARAMS.OLD so that it can be referred to later if needed, and COMP is run to compare NPARAMS with PARAMS. If none of the dimensions in NPARAMS exceed those in PARAMS,TRANS is run and SAFE may be run without any problem; if any dimensions are exceeded, messages are printed out indicating the current dimensions and those required by SAFE.

Two choices are given for what to do next. The first option, which allows the user to stop, gives him an opportunity to re-digitize, if desired. This could also put some control on the changing of dimensions to excessively large numbers. The second choice is to enter an editing routine, which allows individual editing of select items or a replacement of the old file with the new one. If editing of individual items is chosen, PARAMS is modified and NPARAMS is deleted; if a

replacement is desired, PARAMS is deleted and NPARAMS is renamed to be PARAMS. The SAFE programs are then re-compiled and re-established, TRANS is run, and SAFE may be run.

A similar editing routine, LIMEDT, may need to be run for TMNDPT. There are several arrays in TMNDPT which may need to be increased, depending on the complexity of the output. On the CDC, the editing is done by hand, and when SAFE is started up again, TMNDPT is automatically re-compiled. On the Interdata, LIMEDT is automatically run if dimensions, set by PARAMETER statements contained in a file called LIMITS, brought in by a $INCLUDE, are exceeded. Once changes are made, TMNDPT is automatically re-compiled and re-established, and the process can be started up again using USAFE, which will be explained below.

The CDC procedure files used to run SAFE communicate to and from the programs through registers using the CDC IGETR and ISETR functions. The only similar facility available on the Interdata is the END of TASK CODE. Normally, all programs end with an EOT code of Ø, but a call to EXIT can set the code to any value supplied as an argument. The CSS file (procedure file on the Interdata) may then check the code and use the information it obtains to determine the flow of control. This simple setup allows both UAREA and UNPREP to run. But, in UAREA there is a problem. The procedure file is in a loop, but because backward jumps aren't allowed in CSS files, there can be no loop on the Interdata, requiring the entry of a CSS filename to get it going again. The actual looping part of UAREA is contained in UAREA2 so that UAREA2 can be run to start up the loop again.

Running SAFE is slightly more complicated. While UAREA and UNPREP only use one register, SAFE requires three; simply using the EOT code only allows one value to be saved at a time. To solve this problem, a one record long file called REGS is used containing three values, one for each of the three registers. ISETR and IGETR were appropriately changed to read and write values to this file. The programs therefore have complete access to these values, however, since all the CSS file can check is a single EOT code, several intermediate programs had to be written to check a specific one of these values and pass the information on to the CSS file using the code. SAFE also operates in a loop, and there is a CSS file, USAFE, containing the looping portion.

RECURRING CHANGES

The following changes had to be made to many of the programs:

1. The program statements were explicitly for CDC, and had to be commented out.

2. The free format input had to be changed to calls to FREEFM and the free format output had to be changed to calls to FFIOUT, FFROUT, FFSOUT, FIOUTA and FROUTA, depending on number and type, since FORTRAN VII Rev 03 didn't have free format I/O.

3. Calls to EXIT were included along with calls to ISETR to set the END OF TASK CODE as well as the RECS file.

4. Calls to the EOF function were changed to END= for formatted input and to the EOF parameter of FREEFM (the last argument) for free formatted input.

5. A $INCLUDE 9,PARAMS was inserted in every routine requiring the changeable dimensions.

6. Calls were inserted to GRINIT, ALFON and ALFOFF to switch back and forth between the alphanumeric and graphic displays of the HP terminal.

7. All unformatted I/O and manipulation of such files were changed to calls to PIKPAK. This allows the data to be completely packed and 256 bytes records to be used. The resulting file is the same as the CDC version with the addition of some overhead words. This had to be done since long records of unknown length had to be output and other methods created files of extremely large size and a lot of wasted space.

8. A blank had to be added to some of the FORMAT statements for carriage control.

9. Double quotes had to be changed to single quotes.

10. I/O involving LEVELS (TAPE10) were changed from free format to formatted to save time since the formats are well defined.

SPECIFIC CHANGES

What follows is a list of what had to be done to each program. Those changes already explained in the previous section will just be mentioned.

OPTSØ

1. Program statement
2. Free format I/O
3. Call to EXIT

RFPREP

1. Program statement
2. Free format output
3. Formatted TAPE10 reads with END= instead of EOF
4. Removed second READ at the end of the program. Put a statement label on the first read so it could be used for remaining input.

NSPLIT

1. Program statement
2. Free format I/O
3. EOF

BREGNS

1. Program statement
2. Free format I/O
3. EOF

POSTPR

1. Program statement
2. Free format I/O
3. EOF

OPTS1

1. Program statement
2. Free format I/O
3. Call to EXIT

DELETE

1. Changed file name to DDELETE because of a system restriction.
2. Program statement
3. Free format I/O
4. EOF

STRWL

1. Program statement
2. Free format I/O
3. EOF

RDUMP

1. Program statement
2. Free format I/O

OPTS2

1. Program statement
2. Free format I/O
3. EOF
4. Call to EXIT

RENUMBER

1. Program statement
2. Free format I/O
3. EOF

REGION

1. Program statement
2. Free format I/O
3. EOF
4. Wrote out boundary nodes using FORMAT statement 131 instead of 130 so they'd be output one per line since the next program expects them in that format.
5. Carriage control blanks
6. Read in a flag using a FORMAT statement so that a carriage return would be recognized since the free format input routine ignores carriage returns.
7. Changed this flag to JFLAG so as not to confuse it with previous IFLAG.
8. Changed WRITES and FORMATS 130 and 150 to write out PARAMETER statement (and appropriate INTEGER statements) instead of a CDC editor command file.

MAIN

1. Program statement
2. $INCLUDE PARAMS
3. Alpha/Graphic Switch
4. The files associated with logical units 2 and 20 may either be created or used here depending on which option is chosen. Since it is not known till the program is run if new files are needed, the allocation must be handled in the program instead of the CSS file. The CSS file assigns the file and if re-allocation is required, the program closes the files, creates them, and re-assigns them.
5. Free format Input
6. Calls to PIKPAK
7. Call to EXIT

TMNDPT

1. Program statement
2. $INCLUDE PARAMS
3. $INCLUDE LIMITS was inserted to include the PARAMETER statements necessary to re-dimension several changeable arrays.
4. Changed four dimension subscripts from constants to symbolic names to be used in conjunction with the LIMITS file.
5. Calls to PIKPAK
6. Free format Input
7. EOF
8. Set IEXIT to 1 if errors were detected and called EXIT with this value to signal if re-compilation is required or not.

XPATH

1. Program statement
2. $INCLUDE PARAMS
3. Free format I/O
4. Changed the DATA statement initializing BIG to an assignment statement because BIG is a COMMON variable.
5. Calls to PIKPAK
6. EOF
7. Call to EXIT
8. Changed A10 to A4 since that is the maximum number of characters the Interdata 32 bit word can handle. Shortened the maximum length of the header to 20 characters as a result.
9. Changed the calls to MOD to calls of MODD because of a system restriction.

MAIN2

1. Program statement
2. $INCLUDE PARAMS
3. Changed some equivalences so as not to mix integers and floating point numbers. The equivalences were mostly there to save space but some must be there for proper execution of the program.
4. Initialized PNEUT in a loop since the DATA statement wouldn't work with the symbolic name ZMP.
5. Calls to PIKPAK
6. EOF
7. Free format I/O
8. Divided a FORMAT statement onto two lines because the output wouldn't fit on the screen.
9. Calls to EXIT

DPATH

1. Program statment
2. $INCLUDE PARAMS
3. Alpha/Graphic Switch
4. Because of the way the flags are set in MAIN2 this program is only run if paths will actually be displayed.
5. Added a call to a routine called LEVSET which saves the level numbers of the facility so that further on in the program the message indicating that no paths were found on a particular level will be printed only if the level is actually present in the facility. The reason this is necessary is because the actual number of levels may be less than those specified at compilation time since the programs are not recompiled for different facilities unless dimensions are exceeded.

DISPLY

1. Program statement
2. $INCLUDE PARAMS
3. Alpha/Graphic Switch
4. Calls to PIKPAK
5. Changed the Y coordinates of TITLE TT1 from 765 to 760 since it went off the screen on the graphic display of the HP terminal.
6. Replaced the call to FINITT with the actual code of the routine since it contains a STOP but it must be able to continue to switch back to the alphanumeric display.

TRANS

1. Program statement
2. $INCLUDE PARAMS
3. Free format I/O
4. EOF
5. Calls to PIKPAK
6. Carriage Control Blanks

MUPDAT

1. Program statement
2. $INCLUDE PARAMS
3. Formatted reads of LEVELS file
4. EOF
5. Carriage Control Blanks

SAFELIB (overall)

1. $INCLUDE PARAMS
2. Free format I/O
3. Calls to PIKPAK
4. EOF
5. Carriage Control Blanks

SUBROUTINE GETLEV

1. Formatted reads of LEVELS file. Changed these reads to input the right type of data as well.

SUBROUTINE DISPTH

1. Set up the initialization of ILN in a loop because the DATA statement wouldn't work with the symbolic name ZPTH.
2. Alpha/Graphic Switch

SUBROUTINE DISLEV

1. Alpha/Graphic Switch

SUBROUTINE MINDPT

1. Modified EPS and BIG to appropriate values for the Interdata 32 bit word.
2. Changed function calls to the CDC SHIFT function to calls to IROTC which is the Interdata equivalent written for this purpose.
3. Changed .AND. to calls to the IAND function.

SUBROUTINE MOD

1. Changed the name to MODD because of a system restriction.

SUBROUTINE MRNU

1. Added a GO TO at the end so that more than one edit can be done per run.

SUBROUTINE UPDND

1. Split up a line of output so it could fit on the screen.

SUMMRY

1. Program statement
2. Calls to PIKPAK

QUERY

1. Program statement
2. Free format I/O
3. Calls to EXIT

BATLE

1. Program statement
2. EOF
3. Double quotes
4. Free format I/O
5. Split up multiple assignments per line.
6. Changed A10 to groups of A4 and A2. In the cases where lines of alphanumeric data were input and output, this meant adding additional variables.
7. Added a variable, IBRUN, to keep track of whether BATLE is being run for the first time.
8. Call to EXIT
9. Changed 6H to 4H and abbreviated some output.
10. Made variables into arrays in some cases when more than 4 characters were needed.
11. Added calls to a routine called TEMPRD which calls FREEFM the same way many times.
12. Changed SUBROUTINE FORWARD to SUBROUTINE FORWRD and SUBROUTINE SAFEOUT to SUBROUTINE SAFOUT to reduce them down to six characters.
13. Changed .T. to .TRUE. and .F. to .FALSE.

14. Added SUBROUTINE TEMPRD.
15. Changed logical unit 66 to logical unit 8.
16. Changed arrays that were dimensioned based on machine word size.
17. Added calls to a routine, NEWFIL, which re-allocates and assigns a file when a new one is needed since END FILE doesn't work with disk files. Without this, if BATLE is run a second time on the same run, and the second output is shorter than the first, the end of the first will appear at the end of the second since it will still be in the file.

EASI WORKING FROM VAX VERSION

1. Call to IGETR
2. Alpha/Graphic Switch
3. Removal of FONT calls
4. Removal of TBIRD call
5. Abbreviation of 'PROBABILITY' to 'PROBAB' on the Y-axis to enable it to fit on the HP graphic display.
6. The program had to be compiled with the HOLLERITH option to allow the strings in quotes to be passed as arguments.

GCS (Graphics Compatibility System) WORKING FROM VAX VERSION

1. The BLOCK DATA subroutine consisted of 'INCLUDES' to other files. Since the Interdata $INCLUDE is not allowed in BLOCK DATA subroutines, the contents of those files had to be physically included.
2. In SUBROUTINE UDRIN, the common variable TICL was being used as a local variable. To avoid possible confusion, the name of the local variable was changed to TICL2.
3. Since Interdata does not allow initialization of COMMON variables except in BLOCK DATA subroutines, the common variables requiring initialization were put into a BLOCK DATA subroutines located at the end of GCS1.FTN.
4. The computer dependent routines and the I/O routines had to be written.
5. In SUBROUTINE GCSEDT there are two variables that need to be initialized with regard to the word size of the computer used. MAXEXP is the maximum value of an exponent for floating point numbers. JFIELD is the maximum power of 10 which can be represented by an integer. The values used for the Interdata 32 bit word are 10 for JFIELD and 75 for MAXEXP.
6. In SUBROUTINE UPSET IVALU, an integer, and VALUE, a floating point number, were equivalenced. Some of the assignment statements of integers used VALUE instead of IVALU. In particular a change had to be made to use IVALU instead of VALUE in statement 220 to enable EASI to run.

GRAPHIC CHANGES

SAFE has been configured to run on an HP terminal with scrolling and dual display capabilities. The graphic display of the HP terminal when put in compatibility mode, is made to emulate a Tektronix 4010 terminal. However, the resolution is only 720 by 360 whereas the actual Tektronix terminal has a resolution

of 1024 by 780. To resolve this, two modes are available, scaled and unscaled. Scaled mode scales the points to fit into 720 by 360; unscaled allows a 720 by 360 subset to be viewed. In most cases, the scaled mode is sufficient to allow readability. However, for the facility layouts, scaled mode gives a display that is too small to read. Simply converting it to use unscaled mode is not enough because the entire picture needs to be viewed. To resolve this, a different version of the PLOT10 SUBROUTINE RESET is used which sets the variables of the screen coordinates to 720 and 360 so that the transformations done are to the actual screen size of the HP graphic display and not the Tektronix. The resulting output is readable, but because of the nature of the facilities to be used, ships, which are long and narrow, there is a lot of wasted screen space. In the SAFE SUBROUTINE LIMITS, the Y coordinates are modified to proportion them to the X coordinates. If this section of code is removed, the Y coordinates expand to fill the entire screen, giving a somewhat unproportional facility layout, but one with much greater readability. Additionally, in the SAFE SUBROUTINE DISLEV the coordinates used to position the printing of the level numbers in the upper left hand corner of the screen had to be changed from 80, 00 to 56,323 to proportion them for the HP display. The new routines are called RESET2, LIMTS2 and DISLV2. A modified version of DISPTH was created called DISPT2 to call DISLV2 and modified versions of the PLOT10 subroutines INITT and TERM called INITT2 and TERM2 were created to call RESET2. The main programs affected are MAIN and DPATH. Once the modified routines are called and a layout is displayed to the HP terminal the user may obtain a layout to the Tektronix 4054, which is generated by calls to the original subroutines. When the layout is completed control returns to the user's HP terminal. This arrangement allows the user to obtain a layout large enough to read and one that is proportionally accurate.

Chapter 2

CHANGES TO THE SAFE USERS' MANUAL VOL III FOR THE INTERDATA 7/32 VERSION

There are some procedural differences between the Interdata version and the original CDC version of SAFE and therefore are some differences in Volume III of the SAFE Users' Manual which should be noted. The August 1980 publication of the manual will be referred to.

OVERALL DIFFERENCES

SAFE, although comprised of many programs, runs on the CDC 6600 as three continuous procedures without any awareness of when one program ends and another begins. On the Interdata, however, the transitions are more obvious because of the end-of-task-code messages that are printed out each time an individual program ends. There are also a few extra end-of-task-code messages for several intermediate programs that had to be added. Another difference is that while on the CDC machine, the procedures automatically loop to allow continuous runs, the loops must be started up by hand on the Interdata.

TERMINOLOGY

<facility> refers to any sequence of eight letters or numbers, the first character of which must be a letter.

PAGE BY PAGE DESCRIPTION

Section 2.2 p. 29-35

The process of transferring a file from the 4051 to the Interdata is basically the same as transferring it to the CDC 6600 except for a few details. All references to the NOS system should be ignored as well as the information about the 4012 emulator program. The following steps should be used to transfer a file to the Interdata for running SAFE:

1. Insert the initialization tape and enter the following commands:

   ```
   FIND 1
   OLD
   RUN
   ```

2. Remove the initialization tape after the following message appears:
   ---INITIALIZATION COMPLETE---

3. Insert the data tape.

4. Sign on to MTM.

5. Enter FIND n (where is file # of desired file) using the FIND FILE KEY.

6. ENTER COPYA CON:,<filename> where filename is any legal Interdata filename. Typically, for the first level, LEVELØ would be an appropriate filename.

7. Wait for a prompt.

8. Hit the DATA SEND Key. When the data is finished being sent (i.e. the tape's not moving and nothing's being printed out on the screen) hit return.

9. Enter the following commands to delete the first line of the file, which is blank and the last line which is an EOF indicator:

```
EDIT
GET <filename>
T1
If there is a blank line ENTER DE
T :/*:
DE
S*
END
```

10. Go back to step 5 choosing different filenames for each level, until all the levels have been copied.

11. Remove the data tape.

12. Enter EDIT and GET <first *filename*>

13. Enter AP. This will give a line number and a prompt. Just enter a carriage return and it will position itself at the end of the file.

14. Enter INC, <second filename> 1,1- , then INC, <third filename>,1- through INC, <nth filename> 1- for n levels.

15. Enter the command SAVE <facility>.LVL

16. Enter END to get out of the editor.

17. The files containing the individual levels may be deleted since they will no longer be used.

Section 2.3 p. 36 Paragraph 2

To run AREA on the Interdata, type UAREA/G <facility>. When a message indicates to enter UAREA2/G <facility> to continue, the user should do so even

if the STOP option is chosen since the regions are renumbered as part of this option.

Section 2.3 p. 39 Figure 18

The boundary nodes must be entered one per line.

Section 2.3 p. 40 Paragraph 2

Ignore the references to LOCAL file TAPE14.

Section 2.3 p. 42-44

Ignore the references to FILE14 and AUTREG. On the Interdata, a file <facility>.AUT will always be used.

Section 2.3 p. 45 Figure 22

UPREP is referred to as UNPREP and can be invoked by typing UNPREP/G <facility>. The boundary nodes must be entered one per line.

Section 2.4 p. 44-47

All references to the CDC XEDITOR should be ignored. After the regions are generated, another program, COMP, is run to check to see if array dimensions required by the SAFE programs exceed those they have been compiled with. If any do, messages are printed out indicating what values are required. An editing routine is then run which allows individual changes, replacement of all dimensions, or re-digitization. If individual editing is chosen, the user is given the option of editing any of the arrays, or exiting. It will continue to allow editing until the exiting option is chosen. Once the desired array is chosen, the user is prompted for a value, four digits long, right justified, preceded by blanks to re-dimension the array. If many changes were required, the user may chose the replacement option which replaces all dimensions with those calculated, using a file already generated for this purpose. After the chosen option is run, the SAFE programs are automatically re-compiled and re-established. The user must be cautioned that this is a time-consuming operation and is therefore given the opportunity to stop (the re-digitizing option), instead of editing the dimensions.

Section 2.4 p. 48

The following files will be generated in your account and must remain there:

DISPLY.TSK
DPATH.TSK
LEVSET.OBJ
LIMITS
MAIN.TSK
MAIN2.TSK
MUPDAT.TSK
PARAMS
SAFELIB.OBJ
SAFELIB2.OBJ
TEKHP.OBJ
TMNDPT.TSK
TRANS.TSK
XPATH.TSK

The following intermediate may be generated in your account. Most of them will be recreated each time SAFE is run:

PARAMS.OLD	TAPE60
NPARAMS	FIVE
FILE11	TAPEOUT
FILE12	TAPEIN
FILE13	FILE2
TAPE66	BAT2
REGS	BAT3
BAT1	BAT4
EASI.FIL	BAT5
TAPE50	BAT9

The input file used to run SAFE is <facility>.LVL. The following files will be generated for each facility:

<facility>.AUT
<facility>.RDT
<facility>.REG

See Figure 2-1 to see which programs generate the specific files.

Section 3 p. 49

Ignore all references to NOS.

Section 3.1.1 p. 51

All references to NOS and CDC should be ignored. To invoke SAFE type SAFE/G <facility>. When prompted to enter USAFE/G <facility> the user should do so if continuation is desired.

Section 3.1.1 p. 52-53

All nodes must be entered one per line.

Section 3.1.1 p. 54

SAFE is currently configured for implementation on a Hewlett Packard 2648A terminal. After the level number is printed in the upper left corner, the facility layout is finished and a carriage return may be entered to continue. A hardcopy may be made, first, by hitting the green key once and function key eight twice.

Section 3.2 p. 58-60

The editing samples should be ignored. If any of the messages appear, and dimensions need to be increased, an editing routine will automatically be run. It gives the user the option of increasing the dimensions of one of four arrays, or exiting. It will continue to allow editing until the exiting option is finally chosen. Once the desired array is chosen, the user is prompted for a value, four digits long, right justified, preceded by blanks, to re-dimension the array.

Once the changes are made, TMNDPT is re-compiled and re-established. After this is finished, USAFE/G <facility> may be entered again and the user may attempt to run TMNDPT again by specifying Option 3 (running the same pathfinder again).

Section 3.4 p. 79 Paragraph 2

Ignore the information about the 'TIME-LIMIT' message.

Section 4.3 p. 110 Paragraph 1

Running the procedure USAFER on the Interdata means entering USAFE/G <facility>.

Appendix F

Ignore.

Appendix H p. 204-205

Whenever the message 'INPUT 3 NUMBERS' appears and input is required, the first of the three must be specified first on a line by itself. The other two may be specified on the next line, together, separated by commas or blanks.

ADDITIONAL FEATURES

SAFE is designed to run on an HP terminal. The facility layout fills the screen for maximum readibility but as a result is also unproportional. However, a proportional layout may be displayed to the Tektronix 4054. When the HP layout is finished the user is asked if a layout is desired to the 4054. For this layout to appear the 4054 must be removed from MTM and the initialization program must be run. To remove the terminal from MTM type: .MTM REM CRT4: from the system console. For instructions on how to run the initialization program see steps 1 and 2 describing the modifications to p. 29-35. When the 4054 layout is finished, control will return to the user's HP terminal.

POSSIBLE ERRORS

The following are known recoverable problems related to the system which may occur while running SAFE:

1. Buffer error - This occurs if system space has not been set high enough for the current activity on the system. This can be resolved by increasing system space from the system console using the SET SYS command. The current amount of system space can be obtained by typing D M from the system console. If this occurs, the program must be started up from the beginning. The message BUFF-ERR may appear or a LOAD-ERR with TYPE=SPAC.

2. Memory Error - This occurs if there is not enough memory to load a program. The two places this could possibly occur are when running EASIn and DPATH (path display program) which both require approximately 200K. EASI is automatically run for the single path option and if the memory is not available the user must wait. DPATH is only run if the user requests that paths be displayed and could be skipped if memory weren't available. In either case, UAREA2 should be restarted to continue. Information on how much memory is being used can be obtained by typing MAP under MTM of D M from the system console. The message MEM-ERR may appear or a LOAD-ERR with TYPE=MEM.

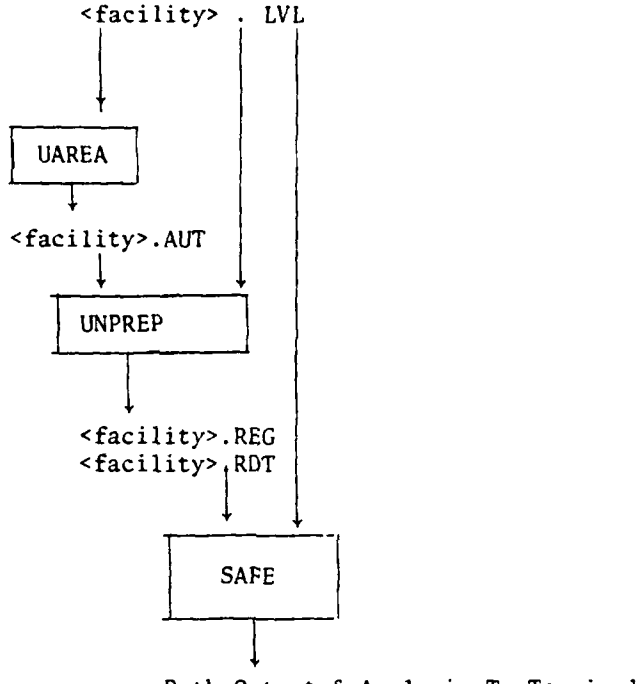


FIGURE 2-1 FILE INPUT AND OUTPUT STRUCTURE OF SAFE ON INTERDATA

DISTRIBUTION

	Copies
Sandia Laboratories, Sandia Corporation Safeguards Methodology Development Division, 4416 P. O. Box 5800 Albuquerque, NM 87185	10
Defense Technical Information Center Cameron Station Alexandria, VA 22314	12

NDW-NSWC(W)-5605/1 (Rev. 5-78)

TO AID IN UPDATING THE DISTRIBUTION LIST FOR NAVAL SURFACE WEAPONS CENTER, WHITE OAK TECHNICAL REPORTS PLEASE COMPLETE THE FORM BELOW:

TO ALL HOLDERS OF NSWC TR 81-321
by Melanie S. Schwartz, Code G42
DO NOT RETURN THIS FORM IF ALL INFORMATION IS CURRENT

A. FACILITY NAME AND ADDRESS *(OLD) (Show Zip Code)*

NEW ADDRESS *(Show Zip Code)*

B. ATTENTION LINE ADDRESSES:

C.

☐ REMOVE THIS FACILITY FROM THE DISTRIBUTION LIST FOR TECHNICAL REPORTS ON THIS SUBJECT.

D.
NUMBER OF COPIES DESIRED _____

GPO 969-784

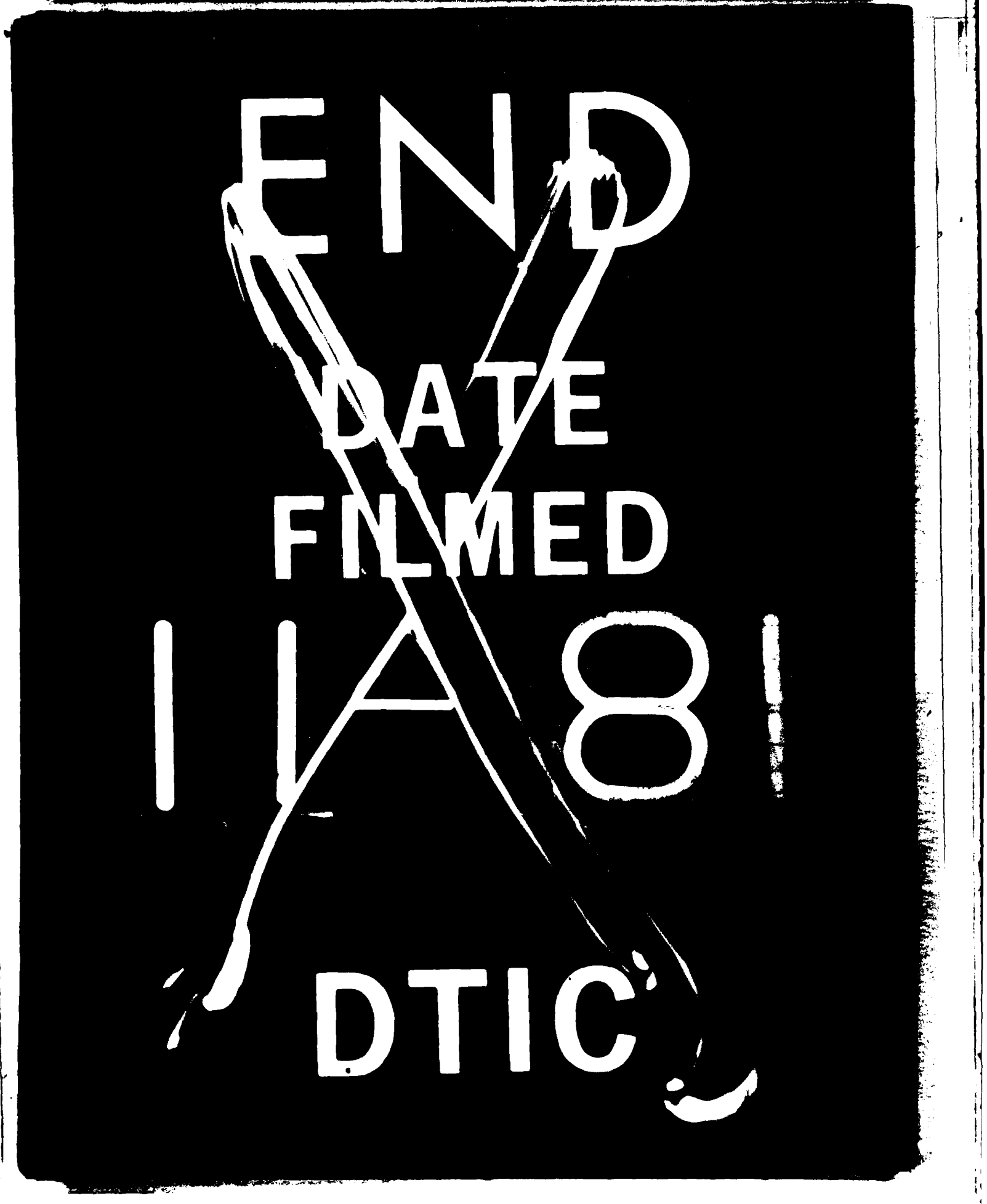
DEPARTMENT OF THE NAVY
NAVAL SURFACE WEAPONS CENTER
WHITE OAK, SILVER SPRING, MD. 20910

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

POSTAGE AND FEES PAID
DEPARTMENT OF THE NAVY
DOD 316

COMMANDER
NAVAL SURFACE WEAPONS CENTER
WHITE OAK, SILVER SPRING, MARYLAND 20910

ATTENTION: CODE G42


END
DATE
FILMED
8
DTIC

AD-A106 261 NAVAL SURFACE WEAPONS CENTER SILVER SPRING MD F/G 9/2
THE SAFEGUARDS AUTOMATED FACILITY EVALUATION (SAFE) ON THE INTE--ETC(U)
AUG 81 M S SCHWARTZ
UNCLASSIFIED NSWC/TR-81-321 NL

END
DATE FILMED
2-82
DTIC

SUPPLEMENTARY

INFORMATION

DEPARTMENT OF THE NAVY
NAVAL SURFACE WEAPONS CENTER
DAHLGREN, VIRGINIA 22448

WHITE OAK LABORATORY
SILVER SPRING, MD. 20910
(202) 394-1800

DAHLGREN LABORATORY
DAHLGREN, VA. 22448
(703) 663-

IN REPLY REFER TO:
E42:AAF:bjj

To all holders of NSWC TR 81-321
Title: The Safeguards Automated Facility Evaluation (SAFE) on the Interdata 7/32

Change 1
1 Dec 1981
6 page (s)

This publication is changed as follows:

Remove the following page and replace with the new page supplied:

iii/iv

Insert the following new pages supplied:

A-1
A-2
B-1
B-2

Insert this change sheet between the cover and the DD Form 1473 in your copy.
Write on the cover "Change 1 inserted"

JOHN M. WACK
By direction

CONTENTS

ILLUSTRATIONS

APPENDIX A

AN UPDATE TO SAFE

INTRODUCTION

SAFE was updated primarily to allow the user to specify probabilities of detection and rates of travel in regions. In addition, there was some cleanup work done to improve the output and fix potential problems.

OVERALL MODIFICATIONS TO INTERDATA VERSION

Due to the acquisition of FORTRAN VII Rev 4, which has free format, there is no longer a need for the subroutines written to handle this function. The calls were removed from all programs except BATLE which contains a substantial amount of free format input. Other than free format, all other changes previously indicated had to be made to the programs. The original Sandia code was used to update the UAREA programs and REGION, and the Interdata versions of the SAFE programs were updated to include Sandia's changes.

UAREA

The following changes were made to the UAREA programs:

1. Calls to EXIT were added.

2. The EOF function was eliminated. It was replaced by END= for file input and for terminal input was ignored since an EOF would only be detected if the user entered a /*. This differs from the CDC in that on that machine carriage returns can be used for EOF.

3. The CDC version was changed to an overlayed configuration in which OPTS1, DELETE, STRWL, RDUMP, and OPTS2 were changed to subroutines. For the Interdata, they were changed back to programs and the original configuration retained. Two new subroutines, RPRDET and RSPEED, were also changed to programs.

REGION

As in UAREA, the free format and EOF were no longer a problem. A change made throughout the programs was that most of the output which was previously free format was formatted. Consequently, output of the boundary nodes didn't have to be modified. All other changes had to be made. One improvement was made. Instead of adding blanks to FORMAT statements, a call was made to CARCON which is a system

routine which allows carriage control to be turned off. Calls to CARCON were added to all the UAREA programs as well.

SAFE

The free format calls were changed back to their original form and the updates made to the Interdata versions of the programs. The changes to be made were determined by comparing the new Sandia code with the last Sandia version. This was done automatically using COMPARE which is a program which compares two files. Changes were made to MAIN, TMNDPT, XPATH, MAIN2, TRANS, MUPDAT, AND SAFELIB. A subroutine, MUPREG, was added to SAFELIB to allow editing of region parameters. TEKHP, which contains the SAFELIB routines which use the HP terminal had to be modified since SUBROUTINE LIMITS was changed. PROGRAM QUERY was modified to use standard free format.

APPENDIX B

INTERNALS OF SAFE

INTRODUCTION

SAFE is configured to use the group account facility of the Interdata operating system. All source code, CSS files and nonchargeable tasks can be found in account 14. Currently, there is one SAFE user, in account 24, which contains data, two files, which enable the programs to have user tailored dimensions (PARAMS and LIMITS), and tasks for changeable programs.

MAINTENANCE

1. UAREA program-Compile and TET using F7CE from account 14. This is a special version of SFCE which allows 20 logical units.

2. REGION-Compile and TET using the same F7CE from account 14.

3. MAIN, TMNDPT, XPATH, MAIN2, DPATH, TRANS and MUPDAT-Compile and TET using SFCEG from account 24.

4. DISPLY-Compile and TET using SFCEGSC from account 24.

5. SUMMRY, QUERY, and BATLES-Compile and TET using SFCE from account 14.

6. EASI-TET using GCSTET from account 14 by typing GCSTET EASI.

USEFUL CSS FILES

1. UAREA.EST-Will compile and TET all the UAREA programs.

2. SAFEST.CSS-Will compile and TET all the changeable SAFE programs from account 24.

3. LIBCMP.CSS-Will compile SAFELIB and create a second copy of the OBJ file, SAFELIB2. The reason this is necessary is because two EDIT's of the OBJ file are necessary for TET and for some reason it could not be rewound for the second pass through. This is done from account 24.

DEBUGGING AID

An important file to dump when problems occur is an intermediate file, generated in MAIN, called TAPEIN. Because it is a binary file written out using

PIKPAK, a program was written, called TAPEIN, to write it out in a meaningful format. To run this program:

1. Recompile and TET to make sure the dimensions are up to date. This must be done from account 24. SFCEG can be used.

2. From account 24 type:

TAPEIN/G